

FORM PTO-1390
(REV 11-98)

U.S. DEPARTMENT OF COMMERCE PATENT AND TRADEMARK OFFICE

ATTORNEY'S DOCKET NUMBER

2000-011

U.S. APPLICATION NO. (If known see 37 CFR 1.5)

09/646796

TRANSMITTAL LETTER TO THE UNITED STATES
DESIGNATED/ELECTED OFFICE (DO/EO/US)
CONCERNING A FILING UNDER 35 U.S.C. 371INTERNATIONAL APPLICATION NO
PCT/GB99/00927INTERNATIONAL FILING DATE
March 24, 1999PRIORITY DATE CLAIMED —
March 24, 1998

TITLE OF INVENTION Apparatus and Method for Providing Transaction Services

APPLICANT(S) FOR DO/EO/US Aravinda Korala

Applicant herewith submits to the United States Designated/Elected Office (DO/EO/US) the following items and other information

1. This is a **FIRST** submission of items concerning a filing under 35 U.S.C. 371.
2. This is a **SECOND** or **SUBSEQUENT** submission of items concerning a filing under 35 U.S.C. 371.
3. This express request to begin national examination procedures (35 U.S.C. 371(f)) at any time rather than delay examination until the expiration of the applicable time limit set in 35 U.S.C. 371(b) and PCT Articles 22 and 39(l).
4. A proper Demand for International Preliminary Examination was made by the 19th month from the earliest claimed priority date.
5. A copy of the International Application as filed (35 U.S.C. 371(c)(2))
 - a. is transmitted herewith (required only if not transmitted by the International Bureau).
 - b. has been transmitted by the International Bureau.
 - c. is not required, as the application was filed in the United States Receiving Ofitce (RO/US).
6. A translation of the International Application into English (35 U.S.C. 371(c)(2)).
7. Amendments to the claims of the International Application under PCT Article 19(35 U.S.C. 371(c)(3))
 - a. are transmitted herewith (required only if not transmitted by the International Bureau).
 - b. have been transmitted by the International Bureau.
 - c. have not been made; however, the time limit for making such amendments has NOT expired
 - d. have not been made and will not be made.
8. A translation of the amendments to the claims under PCT Article 19 (3 5 U.S.C. 37 1 (c)(3)),
9. An oath or declaration of the inventor(s) (35 U.S.C. 371(c)(4)).
10. A translation of the annexes to the International Preliminary Examination Report under PCT Article 36 (35 U.S.C. 371(c)(5)).

Items 11. to 16. below concern document(s) or information included:

11. An Information Disclosure Statement under 37 CFR 1.97 and 1.98.
12. An assignment document for recording. A separate cover sheet in compliance with 37 CFR 3.28 and 3.31 is included.
13. A **FIRST** preliminary amendment.
- A **SECOND** or **SUBSEQUENT** preliminary amendment.
14. A substitute specification.
15. A change of power of attorney and/or address letter.
16. Other items or information:

International Search Report, International Preliminary Examination Report, containing
Copy of amended claims.



23521

PATENT TRADEMARK OFFICE

U.S. APPLICATION NO. (if known - see 37 CFR 1.5)

09/646796

INTERNATIONAL APPLICATION NO.
PCT/GB99/00927ATTORNEY'S DOCKET NUMBER
2000-01117. The following fees are submitted

BASIC NATIONAL FEE (37 CFR 1.492 (a) (1) - (5)):

Neither international preliminary examination fee (37 CFR 1.482)
 nor international search fee (37 CFR 1.445(a)(2)) paid to USPTO
 and International Search Report not prepared by the EPO or JPO \$970.00

International preliminary examination fee (37 CFR 1.482) not paid to
 USPTO but International Search Report -prepared by the EPO or JPO \$840.00

International preliminary examination fee (37 CFR 1.482) not paid to USPTO but
 international search fee (37 CFR 1.445(a)(2)) paid to USPTO \$760.00

International preliminary examination fee paid to USPTO (37 CFR 1.482)
 but all claims did not satisfy provisions of PCT Article 33(I)-(4) \$670.00

International preliminary examination fee paid to USPTO (37 CFR 1.482)
 and all claims satisfied provisions of PCT Article 33(I)-(4) \$96.00

CALCULATIONS PTO USE ONLY

ENTER APPROPRIATE BASIC FEE AMOUNT

\$ 840.00

Surcharge of \$130.00 for furnishing the oath or declaration later than 20 30 months from the earliest claimed priority date (37 CFR 1.492(e)). \$ 0.00

CLAIMS	NUMBER FILED	NUMBER EXTRA	RATE	
Total claims	76 -20 -	56	X \$18.00	\$ 1008.00
Independent claims	5 -3 -	2	X \$78.00	\$ 156.00

MULTIPLE DEPENDENT CLAIM(S) (if applicable) +\$260.00 \$ 0.00

TOTAL OF ABOVE CALCULATIONS \$ 2004.00

Reduction of 1/2 for filing by small entity, if applicable. A Small Entity Statement must also be filed (Note 37 CFR 1.9, 1.27, 1.28). \$ 0.00

SUBTOTAL - \$ 2004.00

Processing fee of \$130.00 for furnishing the English translation later than 20 30 months from the earliest claimed priority date (37 CFR 1.492(t)). + \$ 0.00

TOTAL NATIONAL FEE = \$ 2004.00

Fee for recording the enclosed assignment (37 CFR 1.21 (h)). The assignment must be accompanied by an appropriate cover sheet (37 CFR 3.28, 3.31). \$40.00 per property + \$

TOTAL FEES ENCLOSED - \$ 2004.00

Amount to be: refunded	\$
charged	\$

- a. A check in the amount of \$ 2004.00 to cover the above fees is enclosed.
- b. Please charge my Deposit Account No. _____ in the amount of \$ _____ to cover the above fees. A duplicate copy of this sheet is enclosed.
- c. The Commissioner is hereby authorized to charge any additional fees which may be required, or credit any overpayment to Deposit Account No. 501392 A duplicate copy of this sheet is enclosed.

NOTE: Where an appropriate time limit under 37 CFR 1.494 or 1.495 has not been met, a petition to revive (37 CFR 1.137(a) or (b)) must be filed and granted to restore the application to pending status.

SEND ALL CORRESPONDENCE TO

Shalom Wertsberger
Saltamar Innovations
30 Fern Lane
South Portland, ME 04106
Phone: (207) 799-9733
Fax: (207) 799-9733

**23521**

PATENT TRADEMARK OFFICE

SIGNATURE

Shalom Wertsberger

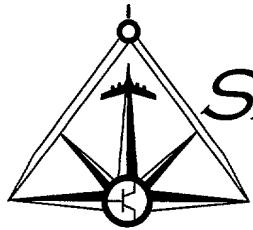
NAME

43,359

REGISTRATION NUMBER

09/646796

529 Rec'd PCT/PTO 22 SEP 2000



**SALTAMAR
INNOVATIONS**

30 Fern lane
South Portland, ME 04106
USA
Phone: +1(207) 799-9733
Fax: +1(207) 799-3698
E-mail: Shalom@Saltamar.com
HTTP://www.saltamar.com

In the United States Patent and Trademark Office

In re application of: **Aravinda Corala**

For: **Apparatus and Method for Providing Transaction Services**

Serial No.:	PCT/GB99/00927	Group:
Filed on:	Even date herewith	Examiner:
Correspondence Date	Sept. 19, 2000	Docket: 2000-011

Preliminary Amendment

Honorable Commissioner of Patents and Trademarks
Washington, D.C. 20231

Sir:

Preliminary to first official action in the above application, currently commencing US national stage under 35 U.S.C. 371, please amend the application as follows:

In the claims:

**Please cancel claims 1-33 as filed and amended in PCT application
PCT/GB99/00927.**

Please enter the following new claims:

34. A method for providing transaction services comprising the steps of:
operating a computer based transaction machine controlled by at least one software application to effect a transaction service;
interacting said software application with a functional interface of a middleware software which extends the functionality of a computer operating system, the computer operating system providing control functions of said computer based transaction machine;
wherein said transaction machine is coupled to at least one transaction device; and

said functional interface provides functionality adapted to the particular hardware of said transaction machine and said transaction device.

35. The method of claim 34 wherein said transaction machine further comprises a data communications interface and wherein said transaction machine is adapted to communicate over said data communications interface.

36. A method for providing transaction services according to claim 34 wherein the transaction machine is selected from a group consisting of an automatic teller machine, an electronic kiosk and an electronic point of sale machine.

37. A method for providing transaction services according to claim 34 wherein said middleware software comprises a series of transaction objects and controls for performing standardized device functions.

38. A method for providing transaction services according to claim 37 wherein said transaction machine further comprises a customizable user interface.

39. A method for providing transaction services according to claim 38 wherein said transaction objects are independent of said user interface.

40. A method for providing transaction services according to claim 39 further comprising a plurality of controls, at least one of which comprises a capabilities interface.

41. A method for providing transaction services according to Claim 40 wherein the capabilities interface can communicate the capabilities of the control.

42. A method for providing transaction services according to claim 37 wherein applications, objects and controls are concurrently operable.

43. A method for providing transaction services according to claim 37 wherein controls are constructed with an event generating capability and wherein a said controls are operable in a selectable mode in which said events are queued up and delivered to an application on demand.

44. A method for providing transaction services according to claim 34 wherein said middleware software is adapted to provide service in accordance with at least one software standard for interacting with different hardware systems.

45. The method for providing transaction services according to claim 44 wherein said at least one software standard is selected from a group consisting of WOSA XFS, OPOS, OFX, TOPEND®, ActiveX®, Javabeans, SNMP.

46. A method for providing transaction services according to claim 34 wherein all errors and transgressions are asserted by the middleware software.

47. A method for providing transaction services according to claim 34 further comprising the step of the middleware software writing trace data to memory and then copies it to disk only when the transaction machine is idle.

48. A method for providing transaction services according to claim 34 further comprising a web browser.

49. A method for providing transaction services according to claim 48 where said at least one software application is operable from within said web browser environment.

50. A method for providing transaction services according to Claim 49 wherein said web browser provides support for software distribution.

51. A method for providing transaction services according to claim 49 further comprising a web browser frame containing at least one device control operable to detect events which must be responded to upon occurrence.

52. A method for providing transaction services according to claim 48 wherein said middleware software comprises a plurality of COM components having a scriptable ActiveX® interface.

53. A method for providing transaction services according to claim 48 wherein said middleware software comprises a plurality of Javabeans™ components having a scriptable interface.

54. A method for providing transaction services according to claim 48 wherein said web browser is adapted to communicate with conventional web sites to be displayed by the computer-based transaction machine.

55. A method for providing transaction services according to claim 48 wherein middleware software allows or disallows access to particular web sites according to a rule database.

56. A method for providing transaction services according to claim 48 wherein middleware software is adapted to customize time-out of the display of individual internet web sites.

57. A method for providing transaction services according to claim 34 wherein the computer-based transaction machine is adapted to allow the software applications and middleware to be altered across a network by an authority.

58. A method for providing transaction services according to claim 34 wherein the transaction machine is adapted to communicate status information to a remote station.

59. The method for providing transaction service of claim 37 wherein said at least one of said transaction objects provide, separately or in combination with other transaction objects and controls, encapsulation of software logic required for performing at least a portion of a transaction.

60. The method of claim 37 wherein at least one of said controls is a device control, providing abstraction of details of a device controlled by said device control.

61. The method of claim 37 further comprising the step of creating a separate thread for each of a plurality of controls.

62. The method of claim 35 further comprising the step of enabling said application program to communicate over said communication interface through a control.

63. The method of claim 37 wherein at least one of said controls implements an OFX interface or a portion thereof, to facilitate communication with an OFX server.

64. The method of claim 34 wherein said middleware software provides generic error handlers.

65. The method of claim 35 further comprising configuring a plurality of transaction machines, and wherein configuration data for said step of configuring is centrally held in a distribution file.

66. The method of claim 38 further comprising the step of constructing said user interface using common web authoring tools.

67. The method of claim 34 wherein said operating system is Microsoft Windows NT.

68. A computer based transaction machine comprising:

at least one transaction device, having a set of capabilities inherent thereto; an operating system to communicate with, and control said transaction device; at least one software application having a user interface, and adapted for execution under control of said operating system; middleware software adapted to interact with said operating system and with said software application, said middleware software having a functional interface adapted to provide an interface to particular hardware capabilities of said transaction machine; and, wherein said middleware software further comprising an application programming interface adapted to provide communication and control services with said transaction device to said software application.

69. The computer based transaction machine of claim 68 wherein said transaction machine further comprises a data communications interface and wherein said transaction machine is adapted to communicate over said data communications interface.
70. A computer based transaction machine according to claim 68 wherein the transaction machine is selected from a group consisting of an automatic teller machine, an electronic kiosk and an electronic point of sale machine.
71. A computer based transaction machine according to claim 68 wherein said middleware software comprises a series of transaction objects and controls for performing standardized device functions.
72. A computer based transaction machine according to claim 71 wherein said transaction machine further comprises a customizable user interface.
73. A computer based transaction machine according to claim 72 wherein said transaction objects are independent of said user interface.

74. A computer based transaction machine according to claim 73 further comprising a plurality of controls, at least one of which comprises a capabilities interface.

75. A computer based transaction machine according to Claim 74 wherein the capabilities interface can communicate the capabilities of the control.

76. A computer based transaction machine according to claim 71 wherein applications, objects and controls are concurrently operable.

77. A computer based transaction machine according to claim 68 wherein controls are constructed with an event generating capability and wherein a said controls are operable in a selectable mode in which said events are queued up and delivered to an application on demand.

78. A computer based transaction machine according to claim 68 wherein said middleware software is adapted to provide service in accordance with at least one software standard for interacting with different hardware systems.

79. The computer based transaction machine according to claim 78 wherein said at least one software standard is selected from a group consisting of WOSA XFS, OPOS, OFX, TOPEND®, ActiveX®, Javabeans, SNMP.

80. A computer based transaction machine according to claim 68 wherein all errors and transgressions are asserted by the middleware software.

81. A computer based transaction machine according to claim 68 further comprising the step of the middleware software writing trace data to memory and then copies it to disk only when the transaction machine is idle.

82. A computer based transaction machine according to claim 68 further comprising a web browser.

83. A computer based transaction machine according to claim 82 wherein said at least one software application is operable from within said web browser environment.

84. A computer based transaction machine according to Claim 83 wherein said web browser provides support for software distribution.

85. A computer based transaction machine according to claim 83 further comprising a web browser frame containing at least one device control operable to detect events which must be responded to upon occurrence.

86. A computer based transaction machine according to claim 82 wherein said middleware software comprises a plurality of COM components having a scriptable ActiveX® interface.

87. A computer based transaction machine according to claim 82 wherein said middleware software comprises a plurality of Javabeans™ components having a scriptable interface.

88. A computer based transaction machine according to claim 82 wherein said web browser is adapted to communicate with conventional web sites to be displayed by the computer-based transaction machine.

89. A method for providing transaction services according to claim 82 wherein middleware software allows or disallows access to particular web sites according to a rule database.

90. A computer based transaction machine according to claim 82 wherein middleware software is adapted to customize time-out of the display of individual internet web sites.

91. A computer based transaction machine according to claim 68 wherein the computer-based transaction machine is adapted to allow the software applications and middleware to be altered across a network by an authority.

92. A computer based transaction machine according to claim 68 wherein the transaction machine is adapted to communicate status information to a remote station.

93. The computer based transaction machine of claim 71 wherein said at least one of said transaction objects provide, separately or in combination with other transaction objects and controls, encapsulation of software logic required for performing at least a portion of a transaction.

94. The computer based transaction machine of claim 71 wherein at least one of said controls is a device control, providing abstraction of details of a device controlled by said device control.

95. The computer based transaction machine of claim 71 further comprising the step of creating a separate thread for each of a plurality of controls.

96. The computer based transaction machine of claim 71 further comprising the step of enabling said application program to communicate over said communication interface through a control.

97. The computer based transaction machine of claim 71 wherein at least one of said controls implements an OFX interface or a portion thereof, to facilitate communication with an OFX server.

98. The computer based transaction machine of claim 68 wherein said middleware software provides generic error handlers.

99. The computer based transaction machine of claim 69 further comprising configuring a plurality of transaction machines, and wherein configuration data for said step of configuring is centrally held in a distribution file.

100. The computer based transaction machine of claim 72 further comprising the step of constructing said user interface using common web authoring tools.

101. The computer based transaction machine of claim 68 wherein said operating system is Microsoft Windows NT.

102. A network comprising a plurality of computer-based transaction machines according to claim 68, one or more networking means and one or more application servers.

103. An Extranet formed by combining a plurality of networks of computer-based transaction machines according to claim 102.

104. An Extranet according to claim 103 provided with a security mechanism which limits the hardware functionality available to individual software applications.

105. A method of providing transaction services comprising the steps of:
operating by a first organization a computer based transaction machine controlled by at least one software application to affect a transaction service;
wherein said software application is provided by a second organization , wherein said software application provides an transaction type different than the transaction type associated with said first organization.

106. A method for selling tickets comprising the steps of:
operating, by a first organization, a computer based automated teller machine having a data communication interface, a display device, an input device, and at least one transaction device adapted for user identification;

executing a software application on said computer based automated teller machine, said software application being adapted to issue tickets for events or services provided by a second organization; and,

automatically charging a user account for said ticket utilizing facilities provided by said automated teller machine.

107. The method of selling tickets of claim 106 wherein said tickets are selected from a list comprising airline tickets, cinema tickets and theatre tickets.

108. A method for selling tickets comprising the steps of:

operating, by a first organization, a computer based kiosk having a data communication interface, a display device, an input device, and at least one transaction device adapted for user identification;

executing a software application on said computer based kiosk, said software application being adapted to issue tickets for events or services provided by a second organization; and,

automatically charging a user account for said ticket utilizing facilities provided by said kiosk.

109. The method of selling tickets of claim 108 wherein said tickets are selected from a list comprising airline tickets, cinema tickets and theatre tickets.

Remarks

Upon entry of PCT application PCT/GB99/00927 into US national stage under 35 USC §371, applicant replaced all claims pending in the application to conform to US practice, and to better point out and more distinctly claim his contribution.

Applicant respectfully requests that the above amendment be entered into record and
that the application be so amended.

Respectfully Submitted



Shalom Wertsberger, Agent for Applicant.

Reg. Num. 43,359
30 Fern Lane
South Portland, ME 04106
Phone: (207) 799-9733; Fax: (207) 799-3698

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number

#3

**STATEMENT CLAIMING SMALL ENTITY STATUS
(37 CFR 1.9(f) & 1.27(b)) -- INDEPENDENT INVENTOR**

Docket Number (Optional)
2000-011
Applicant, Patentee, or Identifier Aravinda KoralaApplication or Patent No. PCT/GB99/00927

Filed or Issued _____

Title Apparatus and Method for Providing Transaction Services

As a below named inventor, I hereby state that I qualify as an independent inventor as defined in 37 CFR 1.9(c) for purposes of paying reduced fees to the Patent and Trademark Office described in

the specification filed herewith with title as listed above.
 the application identified above.
 the patent identified above.

I have not assigned, granted, conveyed, or licensed, and am under no obligation under contract or law to assign, grant, convey, or license, any rights in the invention to any person who would not qualify as an independent inventor under 37 CFR 1.9(c) if that person had made the invention, or to any concern which would not qualify as a small business concern under 37 CFR 1.9(d) or nonprofit organization under 37 CFR 1.9(e).

Each person, concern, or organization to which I have assigned, granted, conveyed, or licensed or am under an obligation under contract or law to assign, grant, convey, or license any rights in the invention is listed below

No such person, concern, or organization exists.
 Each such person, concern, or organization is listed below.

Korala Associates Limited, John Cotton Building, Sunnyside, Edinburgh, EH7 5RA, United Kingdom

Separate statements are required from each named person, concern, or organization having rights to the invention stating their status as small entities. (37 CFR 1.27)

I acknowledge the duty to file, in this application or patent, notification of any change in status resulting in loss of entitlement to small entity status prior to paying, or at the time of paying, the earliest of the issue fee or any maintenance fee due after the date on which status as a small entity is no longer appropriate. (37 CFR 1.28(b))

Aravinda Korala

NAME OF INVENTOR

Signature of inventor

NAME OF INVENTOR

NAME OF INVENTOR

Signature of inventor

Signature of inventor

20 Oct 2000

Date

Date

Date

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

STATEMENT CLAIMING SMALL ENTITY STATUS (37 CFR 1.9(f) & 1.27(c))--SMALL BUSINESS CONCERN

Docket Number (Optional)
2000-011

Applicant, Patentee, or Identifier: Aravinda Korala

Application or Patent No.: _____

Filed or Issued: _____

Title: Apparatus and Method for Providing Transaction Services

I hereby state that I am

the owner of the small business concern identified below:
 an official of the small business concern empowered to act on behalf of the concern identified below:

NAME OF SMALL BUSINESS CONCERN Korala Associates Limited

ADDRESS OF SMALL BUSINESS CONCERN John Cotton Building, Edinburgh, EH7 5RA, United Kingdom

I hereby state that the above identified small business concern qualifies as a small business concern as defined in 13 CFR Part 121 for purposes of paying reduced fees to the United States Patent and Trademark Office. Questions related to size standards for a small business concern may be directed to: Small Business Administration, Size Standards Staff, 409 Third Street, SW, Washington, DC 20416

I hereby state that rights under contract or law have been conveyed to and remain with the small business concern identified above with regard to the invention described in:

the specification filed herewith with title as listed above.
 the application identified above.
 the patent identified above.

If the rights held by the above identified small business concern are not exclusive, each individual, concern, or organization having rights in the invention must file separate statements as to their status as small entities, and no rights to the invention are held by any person, other than the inventor, who would not qualify as an independent inventor under 37 CFR 1.9(c) if that person made the invention, or by any concern which would not qualify as a small business concern under 37 CFR 1.9(d), or a nonprofit organization under 37 CFR 1.9(e).

Each person, concern, or organization having any rights in the invention is listed below:

no such person, concern, or organization exists.
 each such person, concern, or organization is listed below.

More people, concerns, or organizations are listed on a separate sheet

Separate statements are required from each named person or organization having rights to the invention stating their status as small entities. (37 CFR 1.27)

I acknowledge the duty to file, in this application or patent, notification of any change in status resulting in loss of entitlement to small entity status prior to paying, or at the time of paying, the earliest of the issue fee or any maintenance fee due after the date on which status as a small entity is no longer appropriate. (37 CFR 1.28(b))

NAME OF PERSON SIGNING Jim Couser

TITLE OF PERSON IF OTHER THAN OWNER Commercial Manager

ADDRESS OF PERSON SIGNING John Cotton Building, Edinburgh, EH7 5RA, United Kingdom

SIGNATURE Jim Couser

DATE 13.11.00.

2/PRTS

09/646796

1

529 Rec'd PCT/PTO 22 SEP 2000

1 **APPARATUS AND METHOD FOR PROVIDING TRANSACTION SERVICES.**

2

3 The present invention relates to apparatus and a method
4 for providing transaction services. In particular it
5 relates to networked computer-based transaction machines
6 and a method for providing transaction services using
7 said transaction machines.

8

9 Transaction machines are herein defined as any computer-
10 based machine able to interact with a user.

11

12 The term ATM is used herein to refer to any transaction
13 machine able to dispense cash. Typically, such machines
14 can also undertake physical transactions such as
15 inputting information through a keypad or touch screen,
16 making sounds, producing video and printing. They might
17 also be able to read bank cards and such like. Kiosks
18 are transaction machines unable to dispense cash, but
19 otherwise able to provide a range of interactive
20 features, often relating to financial services. For test
21 purposes, a conventional PC may be used as a transaction
22 machine.

23

24 Electronic cash machines are a large and rapidly growing
25 market. Many different hardware providers produce

1 equipment for this market such as the machines
2 themselves, the servers to which they connect and the
3 networking means through which they typically
4 communicate. Furthermore, many different operating
5 systems and applications are used both for operating and
6 developing these systems.

7

8 As a result of the complexity and diversity of hardware
9 and software currently being used in this field, it is
10 difficult and expensive to alter these systems to extend
11 their functionality, upgrade to newer and better
12 hardware, software or networking means or to interface
13 with other systems. As it is difficult to make even
14 small changes to complex systems without running the risk
15 of their malfunctioning, the evolution of such systems is
16 slow.

17

18 It would therefore be advantageous to find a way of
19 making it easier to alter the hardware, software and
20 network components of ATMs/kiosks, their servers and
21 their networking means.

22

23 Furthermore, it would be advantageous to provide a means
24 for enabling such changes to be implemented in small
25 stages.

26

27 Yet further, it would be advantageous to find a way to
28 reduce the risk of such systems malfunctioning.

29

30 In current practice, it is difficult and therefore
31 expensive to operate ATM/kiosk networks containing
32 diverse hardware, software and networking means. Often
33 large amounts of hardware and software must be upgraded
34 concomitantly to reduce interface problems. Furthermore,
35 it is difficult to interface networks of dissimilar

1 devices, perhaps belonging to different organisations.
2 If dissimilar ATM/kiosk systems could be readily
3 interfaced, forming a so-called Extranet, new and useful
4 co-operative applications could be developed which,
5 although currently possible, are prohibitively complex
6 and expensive at the present time.

7

8 It would therefore be advantageous to provide a better
9 means of networking ATMs/kiosks which use diverse
10 hardware, software and networking implementations. In
11 particular, it would be advantageous to provide a means
12 of allowing co-operation between dissimilar networks.
13 Furthermore, it would be advantageous to reduce the
14 amount of work required to enable ATM/kiosk applications
15 to run on dissimilar hardware implementations.

16

17 At the present time, there is a rapid growth in
18 electronic commerce (e-commerce), usually conducted over
19 the internet. E-commerce is being limited by
20 difficulties gaining access to the internet for many
21 consumers and due to the limitations of the machines
22 currently used by consumers for internet transactions. A
23 typical e-commerce consumer will access a web site using
24 a home PC. However, home PCs lack facilities such as the
25 ability to dispense cash or read a smartcard which are
26 important in many types of common financial transaction.

27

28 It would therefore be desirable to provide a means of
29 allowing internet-based e-commerce to be accessed from
30 ATMs and kiosks which already have hardware facilities
31 suitable for financial transactions. This would allow e-
32 commerce services to be provided which required expensive
33 or high-security hardware facilities which cannot be
34 securely provided at a reasonable cost on privately owned
35 web browsers. Furthermore, it would be possible for e-

1 commerce to be made readily available to a much larger
2 base of consumers than is currently available.

3

4 The design of ATM networks typically involves input from
5 numerous professionals such as software and hardware
6 engineers specialising in the various systems,
7 applications and communications means, graphics and GUI
8 specialists, language specialists and so forth. In
9 current working practice these specialists are highly
10 dependent on each other and much time and money is spent
11 communicating different requirements amongst people
12 working on diverse areas of a project.

13

14 It would therefore be advantageous to provide a means by
15 which the different specialists working on a project may
16 work more independently. In particular, it would be
17 highly advantageous to provide a means by which the
18 different specialists may customise elements of the
19 application pertaining to their own specialisation
20 without affecting other elements of the application. It
21 would be particularly advantageous if the different
22 specialists were able to use well known prior art
23 authoring tools to prepare aspects of the application.
24 According to the present invention there is provided a
25 method for providing transaction services wherein

26

27 (a) the user of the transaction services interacts
28 with a computer-based transaction machine which is
29 controlled by one or more software applications;

30

31 (b) the software applications interact with the
32 functional interfaces of middleware software, which
33 extends the functionality of an underlying operating
34 system; and

35

1 (c) said functional interfaces provide functionality
2 which is implemented in a manner adapted to the
3 particular hardware capabilities of the transaction
4 machine.

5

6 The computer-based transaction machine may be selected
7 from a group which comprises automatic teller machines,
8 kiosks, electronic point of sale machines and the like.

9

10 Preferably, the middleware software comprises a series of
11 transaction objects and controls for standard device
12 functions.

13

14 More preferably, transaction objects are independent of
15 the interface between the user and the transaction
16 machine; the interface between the user and the
17 transaction machine being customisable.

18

19 Preferably, the controls implement a capabilities
20 interface.

21

22 More preferably, the capabilities interface is able to
23 communicate the capabilities of the control software.

24

25 The applications, objects and controls may be fully
26 concurrent and asynchronous.

27

28 The controls may have a mode in which events are queued
29 up and delivered to the application on demand.

30

31 Preferably, controls can run on the transaction machine
32 even when supported hardware devices are not present.

33

1 More preferably, the middleware software uses one or more
2 open standards for interacting with different hardware
3 systems.

4

5 Preferably, the middleware software only provides
6 cancellation commands for functions which can be
7 successfully cancelled.

8

9 The middleware software may only requires a timeout
10 command to be supplied when it is meaningful to do so.

11

12 Preferably, all controls are persistent.

13

14 More preferably, there is provided a control containing a
15 persistent object.

16

17 Preferably, all errors and transgressions are asserted by
18 the middleware software.

19

20 Preferably, the middleware software provides a trace
21 facility that is always enabled and which logs trace
22 events.

23

24 The middleware software may use a ring buffer to store a
25 log of trace events.

26

27 Preferably, the middleware software writes trace data to
28 memory and then copies it to disk only when the
29 transaction machine is idle.

30

31 Preferably, one or more software applications are hosted
32 in a web browser.

33

1 More preferably, the use of a web browser provides
2 support for software distribution and network
3 connections.

4

5 An additional browser frame may be provided which
6 contains the device controls required to detect events
7 which must be dealt with immediately they occur.

8

9 The middleware software may comprise a series of COM
10 components with a scriptable ActiveX® interface.

11

12 The middleware software may comprise a series of
13 Javabeans™ components with a scriptable interface.

14

15 The use of a web browser may allow conventional web sites
16 to be displayed by the computer-based transaction
17 machine.

18

19 Preferably, the middleware software allows or disallows
20 access to particular web sites according to a rule
21 database.

22

23 The middleware software may be adapted to customise time-
24 out of the display of individual internet web sites.

25

26 Preferably, said computer-based transaction machine is
27 adapted to allow the software applications and middleware
28 to be altered across a network by an authority.

29

30 More preferably, the transaction machine communicates
31 information about its status to a remote monitoring
32 station across a network.

33

34 According to a second aspect of the present invention,
35 there is provided a computer-based transaction machine;

1 wherein said computer-based transaction machine is
2 provided with hardware devices for interaction with users
3 and the exchange of transaction-related information with
4 other machines; wherein said computer-based transaction
5 machine is controlled by one or more software
6 applications; wherein said software applications control
7 hardware devices through functional interfaces with
8 middleware software; wherein said middleware software
9 extends the functionality of an underlying operating
10 system and wherein said functional interfaces are
11 hardware independent but provide functionality which is
12 implemented in a manner adapted to the capabilities of
13 the particular hardware devices which are provided.

14

15 The computer-based transaction machine may be selected
16 from a group which comprises automatic teller machines,
17 kiosks, electronic point of sale machines and the like.

18

19 Preferably, the middleware software comprises a series of
20 transaction objects and controls for standard device
21 functions.

22

23 More preferably, transaction objects are independent of
24 the interface between the user and the transaction
25 machine; the interface between the user and the
26 transaction machine being customisable.

27

28 Preferably, the controls implement a capabilities
29 interface.

30

31 More preferably, the capabilities interface is able to
32 communicate the capabilities of the control software.

33

34 The applications, objects and controls may be fully
35 concurrent and asynchronous.

1 The controls may have a mode in which events are queued
2 up and delivered to the application on demand.
3
4 Preferably, controls can run on a transaction machine
5 even when supported hardware devices are not present.
6
7 More preferably, the middleware software uses one or more
8 open standards for interacting with different hardware
9 systems.
10
11
12 Preferably, the middleware software only provides
13 cancellation commands for functions which can be
14 successfully cancelled.
15
16 The middleware software may only requires a timeout
17 command to be supplied when it is meaningful to do so.
18
19 Preferably, all controls are persistent.
20
21 More preferably, there is provided a control containing a
22 persistent object.
23
24 Preferably, all errors and transgressions are asserted by
25 the middleware software.
26
27 Preferably, the middleware software provides a trace
28 facility that is always enabled and which logs trace
29 events.
30
31 The middleware software may use a ring buffer to store a
32 log of trace events.
33

1 Preferably, the middleware software writes trace data to
2 memory and then copies it to disk only when the
3 transaction machine is idle.

4

5 Preferably, one or more software applications are hosted
6 in a web browser.

7

8 More preferably, the use of a web browser provides
9 support for software distribution and network
10 connections.

11

12 An additional browser frame may be provided which
13 contains the device controls required to detect events
14 which must be dealt with immediately they occur.

15

16 The middleware software may comprise a series of COM
17 components with a scriptable ActiveX® interface.

18

19 The middleware software may comprise a series of
20 Javabeans™ components with a scriptable interface.

21

22 The use of a web browser may allow conventional web sites
23 to be displayed by the computer-based transaction
24 machine.

25

26 Preferably, the middleware software allows or disallows
27 access to particular web sites according to a rule
28 database.

29

30 The middleware software may be adapted to customise time-
31 out of the display of individual internet web sites.

32

33 Preferably, the computer-based transaction machine is
34 adapted to allow the software applications and middleware
35 to be altered across a network by an authority.

1

2 More preferably, the transaction machine can communicate
3 information about their status to a remote monitoring
4 station across a network.

5

6 According to a third aspect of the present invention
7 there is provided a network comprising a plurality of
8 computer-based transaction machines, one or more
9 networking means and one or more application servers.

10

11 According to a fourth aspect of the present invention,
12 there is provided an Extranet formed by combining a
13 plurality of networks of computer-based transaction
14 machines.

15

16 Preferably, the Extranet is provided with a security
17 mechanism which limits the hardware functionality
18 available to individual software applications.

1 An example embodiment of the present invention, referred
2 to as the system, will now be described with reference to
3 the following Figures wherein:

4

5 Figure 1 shows a simple ATM network;
6 Figure 2 shows an ATM network with diverse hardware;
7 Figure 3 shows two distinct networks being combined
8 to form an Extranet; and
9 Figure 4 shows the software architecture of the
10 preferred implementation of the system.

11

12 Figure 1 shows a simple ATM network comprising a server
13 1, a networking means 2 and an ATM 3. The system is
14 designed to operate such networks and also more complex
15 networks such as shown in Figure 2 wherein there may be
16 ATMs of different functionality, here labelled 4.

17

18 A particular benefit of the system is its ability to
19 allow distinct networks to operate together as shown in
20 Figure 3. Here, two distinct networks 5 and 6 operated
21 by distinct servers 7 and 8 are connected 9. The
22 resulting joined network is referred to as an Extranet.

23

24 By joining multiple networks together, it becomes
25 possible for different organisations to co-operate in the
26 provision of ATM/kiosk network services. For example,
27 suppose that a bank which owned a series of conventional
28 ATMs and an airline which owned a series of ticketing
29 kiosks chose to co-operate. There exists the potential
30 for the bank's ATMs to both allow customers to pay for an
31 airline ticket and to print out that ticket. Similarly,
32 the airline might offer a limited selection of banking
33 services, such as balance display, which are compatible
34 with the functionality of their kiosks.

35

1 Using prior art, the development of such a system would
2 be complex, particularly due to the different hardware
3 and capabilities of the bank's ATMs and the airline's
4 kiosks. Such co-operation between organisations is by no
5 means impossible at the present time, but is currently
6 rare due to the complexity and expense required for
7 implementation.

8

9 In general, the system provides a means for a plurality
10 of servers to operate a plurality of ATMs and kiosks
11 using a plurality of networking means. An example
12 application would be to allow consumers to purchase eg
13 cinema, theatre and airline tickets from different
14 organisations through ATMs positioned at convenient
15 locations.

16

17 Typically, the networking means will be the internet, a
18 corporate intranet or LAN but may be any networking means
19 or a mixture of networking means.

20

21 The system comprises a middleware software layer which
22 extends the function of an underlying operating system
23 and which in turn provides a single programming interface
24 for an ATM/kiosk control application to be written to.

25

26 Figure 4 shows the software architecture of the preferred
27 implementation of the system. An ATM/kiosk control
28 application 10 is hosted in a web browser 11 such as
29 Microsoft®'s Internet Explorer. The application runs on a
30 computer with a particular operating system, 12, such as
31 Windows NT®, the functionality of which has been extended
32 by middleware software 13.

33

34 The middleware comprises a series of components and
35 objects, for use by the application, which extend the

1 functionality of the operating system and provide tools
2 to simplify development of the ATM application.

3

4 In the preferred implementation all of the system's sub-
5 systems are implemented as a series of COM components
6 with an ActiveX® interface or as Javabeans™ with a
7 scriptable interface. This architecture enables
8 applications running within Internet Explorer to access
9 functionality provided by the operating system and the
10 middleware, including access to hardware.

11

12 A useful benefit of this implementation is that
13 applications may be prepared using common authoring tools
14 and such as Microsoft®'s FrontPage®, VisualStudio®, Visual
15 Interdev® and common development environments such as
16 Visual Basic®, Visual C++®, Powerbuilder®, Delphi® etc.
17 This means that applications can be prepared with tools
18 with which developers will be familiar and which, due to
19 their popularity, provide facilities and support that
20 would be prohibitively expensive to prepare for a custom
21 development environment.

22

23 A further benefit of using browser technology is that
24 they provide an environment in which software download
25 can be readily controlled. The application may be held
26 entirely locally to an ATM/kiosk, entirely on a server or
27 any compromise between these two extremes. The
28 application can be downloaded daily if required.

29

30 The system uses the Windows® Open System Architecture
31 Extensions for Financial Services (WOSA XFS) to support
32 ATM hardware in a vendor independent manner.

33

34 The system also uses the Object Linking and Embedding for
35 Point Of Sale (OPOS) standard for interacting with

1 different hardware systems. This means that applications
2 can access hardware independent of whether the underlying
3 hardware supports WOSA XFS or OPOS.

4

5 The system also supports the PC/SC standard for
6 smartcards, thereby providing a uniform way of accessing
7 smartcards.

8

9 Furthermore, the system also provides support for a
10 variety of other open standards such as OFX and SNMP and
11 transaction monitors such as NCR's TOPEND®.

12

13 Clearly, support for additional standards may readily be
14 added.

15

16 The primary subsystems of the middleware software
17 comprise a series of wizards, device controls, self-
18 service controls, communications controls and status
19 monitoring components.

20

21 The top level components are the wizards, which are a
22 series of transaction objects that implement common
23 ATM/kiosk transactions such as dispensing cash, printing
24 a statement etc. In the preferred embodiment, each is
25 implemented as an ActiveX® object or a Javabean™. Whilst
26 wizards are running, they take control of the function of
27 the ATM/kiosk. Wizards interface with other controls and
28 encode all of the top-level control logic.

29

30 Applications can be built with the system by customising
31 and combining wizards. Wizards encapsulate all of the
32 features and functionality required by a particular
33 transaction or chunk of application. When using ActiveX®,
34 Wizards receive input via ActiveX® properties and methods
35 and output their state as a set of ActiveX® events.

1 Alternatively the wizard can be implemented in the same
2 way as a Javabean™. As a result of this design feature,
3 the wizard is completely independent of the ATM/kiosk-
4 user interface.

5

6 For example, an ATM might have a single button which
7 dispenses \$10 on demand. A second ATM might implement
8 more complex controls and display a detailed animation
9 whilst money is issued. However, the same wizard may be
10 used to implement both these ATMs. The wizard
11 encapsulates the essential software logic of the
12 transaction while allowing the user interface to be
13 freely defined by script on the browser page.

14

15 This has several important benefits which will lead to
16 time and cost savings: firstly, the encapsulated features
17 within the wizard can be reused between different
18 applications whilst allowing the different applications
19 to have totally different look and feel. Secondly, this
20 allows the user interface to be designed with common web
21 tools. Thirdly, the user interface may be designed
22 without any risk of compromising the function of the
23 wizard. Finally, the user interface may be designed by a
24 specialist who may not be an expert in the other aspects
25 of ATM/kiosk software and hardware.

26

27 An additional important feature of the wizards is that
28 they are able to interpret the capabilities of the
29 hardware on which they are run. For example, they may be
30 able to establish whether a cash dispensing means is
31 available. One application may then run on a plurality
32 of different hardware implementations, adapting its
33 functionality to the capabilities of that hardware.

34

1 This not only allows different hardware implementations
2 to be incorporated into the same network but allows
3 distinct networks to be joined into an Extranet.

4

5 The device controls provide hardware independent access
6 to the special devices on an ATM or kiosk. Each device
7 control acts as a persistent server that can be
8 controlled and interrogated by one or more applications
9 or wizards. A device control abstracts the details of
10 the hardware underneath it and acts as a complete server
11 for that device. Applications and wizards interact with
12 controls through a scriptable ActiveX® interface or a
13 Javabeans™ interface.

14

15 Some example device controls supported by the system are:

- 16 • Camera
- 17 • Card Reader (motorized, swipe, DIP, smart cards etc.)
- 18 • Cash Acceptor
- 19 • Cash Dispenser
- 20 • Coin Dispenser
- 21 • Depository
- 22 • Doors
- 23 • Encryptor
- 24 • Guide Lights
- 25 • Indicators
- 26 • Journal Printer
- 27 • Keyboards
- 28 • Laser Printers
- 29 • Modems
- 30 • Operator Panel
- 31 • Passbook (including page turn)
- 32 • Pin Pad
- 33 • Receipt Printer

- 1 • Scanner
- 2 • Sensors
- 3 • Signature Capture
- 4 • Statement Printer
- 5 • Touchscreen
- 6 • UPS
- 7 • VendorMode
- 8 • Weighing Scales

9
10 Multiple applications may be run simultaneously and
11 device controls are fully concurrent. This is important
12 as the cycle time of ATMs and kiosk transactions can be
13 critical. Their design is such that they can be used in
14 an event-driven manner, with controls reporting their
15 result (success or failure) via ActiveX® or Javabeans™
16 events, or in a procedural manner from within a language
17 such as C++. In the event-driven mode, applications can
18 be readily created using browser technology; for example,
19 readily available web tools which provide appropriate
20 easy-to-use graphical interfaces can be used to create
21 event-driven applications.

22
23 In order to be able to operate asynchronously, all
24 controls create their own thread, called the event
25 thread, when first constructed. When an asynchronous
26 method is called, a command message is sent to the event
27 thread. The event thread carries out the command and
28 sends a message back to the main thread on completion:
29 the completion method causes the appropriate event to be
30 fired. By implementing commands using the event thread,
31 the main application thread is free to process other
32 tasks in parallel. The event thread also ensures that
33 the device states persist from one application page to
34 another: although controls on browser pages are being

1 continually created and destroyed, the event thread
2 remains running and ensures that the connection to the
3 device is never lost.

4

5 When controls are run in a procedural manner, from a
6 language such as C++, the controls may be set to a mode
7 in which events are queued up and delivered to the
8 application on demand, allowing the application to carry
9 out other tasks, and return to the event queue at an
10 appropriate time.

11

12 The self-service controls provide the functionality
13 necessary for creating self-service applications.

14 Important self-service controls are described further
15 below. The communications controls provide access to the
16 remote host computers. Both the self-service and
17 communications controls have the same server architecture
18 as the device controls and all may be executed
19 asynchronously.

20

21 The status monitoring system monitors the health of the
22 ATM or Kiosk and sends status and alert signals to an
23 external monitoring station using SNMP alerts.

24

25 All controls implement a capabilities interface, allowing
26 an application or wizard to interrogate the capabilities
27 of the control as well as the device which the control
28 represents.

29

30 Therefore, not only can different hardware
31 implementations be integrated into the same network or
32 Extranet, the applications can dynamically configure the
33 services they provide depending on the capabilities of
34 the hardware available on the kiosk.

35

1 As a result of this design, individual software
2 components can be upgraded without having to change other
3 aspects of the application. New features can be added
4 without making the application dependent on those
5 features.

6
7 Furthermore, hardware and networking components may be
8 upgraded or altered step by step. Due to the modular
9 nature of the system and its customisability, a plurality
10 of communications and hardware implementations may be
11 used at once. This means that an organisation which runs
12 an ATM/kiosk network might use its legacy communications
13 and hardware implementations, perhaps concurrently with
14 Internet/Intranet support. This means that ATM networks
15 may be implemented and altered step-wise.

16
17 Such upgrades are particularly easy when using the Open
18 Financial Exchange (OFX) architecture. The middleware
19 software implements a single OFX Control which may
20 interface with an OFX server by any networking means.
21 The OFX server may also interface with a host by any
22 networking means. Once this architecture is implemented,
23 the resulting network topology may be readily altered,
24 making this an easy migration path for existing networks
25 to use this system.

26
27 A further implication of the design of the controls is
28 that they can run on an ATM/kiosk even when actual
29 hardware devices are not present. This allows the
30 applications to be started up and run, for example for
31 development and test purposes, without requiring
32 particular hardware. When the application requests the
33 capabilities of a particular control, the control will
34 reply that the device is not present and that the
35 capabilities are null. Therefore it is possible to

1 create and test application on, for example, a PC. In
2 this situation, the PC will behave like an ATM/kiosk in
3 its interactions with the application.

4

5 An ignore mode is also provided wherein particular
6 controls will return "success" for every command. This
7 allows the application to use generic code which does not
8 need to test whether the device is present at each step,
9 simplifying the code that needs to be written when
10 creating an application to cope with various hardware
11 capabilities.

12

13 An HTML-based application is also provided with the
14 system for testing device controls. This application
15 allows the operator to select a subset of the devices for
16 testing. For each device, two test sequences are
17 defined: one requires operator interaction (e.g.
18 entering/removing a card) and one requires no operator
19 interaction. When the latter is selected, the
20 interaction-free test sequences will be repetitively run
21 for the selected devices, allowing applications provided
22 using this system to be easily stress tested. Complete
23 tests including operator interaction may also be
24 selected. Testing is automated and therefore as
25 reproducible as possible.

26

27 All controls include a security mechanism. This
28 mechanism allows the methods of the various controls to
29 be enabled and disabled. This is particularly important
30 in an Extranet environment when applications of differing
31 abilities run on a given kiosk or ATM. For example, if a
32 bank operating a network of ATMs allowed an airline to
33 dispense tickets through its ATMs by way of an Extranet,
34 it would wish to disallow the airline's application from
35 dispensing cash.

1

2 This security mechanism is implemented by a key passing
3 technique as follows:

4

5 The middleware software contains a security control which
6 allows the current security configuration of an ATM or
7 kiosk to be set. Using the security control, the owner
8 of the ATM or kiosk can specify details of the security
9 configuration (i.e. which methods of a control are
10 allowed and disallowed). Applications identify
11 themselves to the security control via a digital
12 certificate which sets the security configuration as
13 specified by the ATM/kiosk owner. If the application
14 attempts to call a disallowed method of control, a trap
15 is generated, transferring control to the ATM/kiosk
16 owner's application.

17

18 An important benefit of the system is that it may readily
19 be used to provide internet based e-commerce facilities
20 through ATMs and kiosks, not only allowing e-commerce
21 facilities to be used by a larger consumer base but also
22 enabling e-commerce which requires expensive or high-
23 security hardware facilities such as cash dispensers or
24 identity verification means that cannot readily be
25 provided on privately owned PCs and web-browsers.

26

27 To help enable this, the system provides a Site-Minder
28 control which allows existing web sites to be safely
29 delivered via ATMs and kiosks. This control provides
30 several important features. For example, it monitors the
31 URL of each page of the web-site being delivered and
32 allows or disallows the page according to a rules
33 database. This stops the user from straying into other
34 web-sites or web-pages that are not normally part of the
35 purpose of the ATM/kiosk. The control allows each page

1 to be given a customised time-out which is important as
2 web sites are normally designed for use at home and have
3 different (longer) time-outs than would be appropriate
4 for public ATMs/kiosks. Web pages may be navigated using
5 a touch sensitive screen, making them intuitive and easy
6 to use. The control can also magnify small features on a
7 web page (such as hypertext links and images with links)
8 This magnification can be toggled on and off by the user,
9 thereby animating the hypertext link. This is beneficial
10 firstly because it makes it easier for the user to see
11 where the link is and secondly because it becomes easier
12 for the user to select the link when it is in its
13 magnified state.

14

15 An additional feature provided by the system for use with
16 ATMs/kiosks with touchscreens is a "softkeyboard" wherein
17 a keyboard is displayed on the touch screen and contact
18 with the displayed keyboard is interpreted by the system
19 like keystrokes on a real keyboard, thereby removing the
20 need for a physical keyboard to be provided.

21

22 One problem commonly faced by web designers is that
23 objects placed on a web page are destroyed when the page
24 is changed. A useful benefit of the middleware is that
25 the ActiveX® hook idea solves this problem - underlying
26 objects remain persistent while lightweight hooks on each
27 page access the object.

28

29 Lack of persistence also leads to problems for the
30 application developer in storing application-wide data.
31 A solution to this problem is provided by a scratchpad
32 control which has a persistent object at its core and
33 allows the application to store and retrieve data at any
34 time. This control supports the Vbscript variant type,
35 allowing all types of data to be stored and retrieved.

1 Furthermore, this control allows data to be shared
2 between multiple applications, marking it as shared.

3
4 A related problem when implementing web-based ATM
5 applications relates to events which must be dealt with
6 immediately, no matter when the event occurs. For
7 instance, if a safe door is opened, an application may
8 need to shut down immediately. This would not be easy to
9 implement in a web-based environment as every page would
10 have to contain some code to handle the event. This
11 problem can be solved in the system by operating a
12 second, invisible frame alongside the main application
13 frame. The invisible frame contains all the device
14 controls needed to detect the events that must be reacted
15 to. This frame may then take control, perhaps closing
16 down the main frame.

17
18 Error handling in traditional ATM applications is
19 difficult. Components may return a large number of error
20 cases, resulting in complex code.

21
22 The middleware software separates the responses it sends
23 to the application into "good responses" and error
24 responses. Most commands have a single good response and
25 all errors are mapped to a single error response,
26 although some may have a plurality of good responses.
27 Good responses allow the application to continue. When
28 an error response is returned, the current transaction
29 flow is normally aborted and control flow jumps out of
30 the normal flow process to handle the error situation.
31 The application can then interrogate the control to
32 determine the exact cause of the error.

33
34 A benefit of this approach is that normal flow is not
35 cluttered by handlers for each of the error cases which

1 can occur. Control may be transferred to generic error
2 handlers which can either recover from the error or abort
3 the transaction completely, perhaps even rebooting the
4 ATM/kiosk. Application code can therefore remain as
5 clear and concise as possible whilst encouraging the
6 application developer to handle all error cases by
7 calling an error handler. In the development
8 environment, fatal errors result in a message box being
9 displayed. A single type of event, DeviceError, is
10 generated when there is some kind of hardware failure,
11 allowing error handling for hardware failure to be
12 encapsulated rather than scattered over many error
13 handlers.

14

15 The system requires applications to interact with it in a
16 well defined way. Even small transgressions are detected
17 and error responses generated; when this happens, the
18 current environment is abandoned and the application is
19 terminated.

20

21 This is based on the well known software engineering
22 approach of assertion; however, the system's assertion
23 differs from common practice by asserting absolutely all
24 disallowed cases, whether serious or not. As a result of
25 this strategy of escalating errors to maximum
26 seriousness, errors are found earlier at development time
27 or at system test time and never allowed to reach a live
28 environment. Although there is a risk of the application
29 reporting a fatal error in the field for a relatively
30 minor problem, this strategy achieves a particularly high
31 level of robustness in comparison to prior art software
32 applications.

33

34 An additional error-handling feature is provided by the
35 way in which the system deals with tracing. In software

1 engineering, tracing is typically enabled only when a
2 problem is suspected; however, this can affect the
3 dynamics of a program, making it harder to find bugs.
4 This is a particularly substantial problem when dealing
5 with time-critical ATM/kiosk applications. However, if
6 conventional tracing was simply always enabled throughout
7 both development and operation of the ATM/kiosk, there
8 would be both performance problems due to, for example,
9 the time spent writing to a hard drive and large quantity
10 of disk space required to store the large number of trace
11 events that will typically be produced.

12

13 The middleware software provides a trace control which
14 records all trace events of the application and
15 underlying middleware and is always enabled. Performance
16 problems are dealt with by writing trace data to memory
17 and writing to disk only when the ATM/kiosk is idle.
18 Cash-dispensing machines and kiosks go through an idle
19 cycle between two users which provides sufficient time to
20 write to disk, even when people are queuing at the
21 machine. Disk space problems are eliminated by using a
22 ring buffer of fixed file size, allocated at boot-up and
23 constant in size throughout operation. When the buffer
24 is full, the oldest data is overwritten, thereby leaving
25 a continual record of the most recent events.

26

27 As a result of this tracing strategy it is much easier to
28 understand one-off or rare problems, which is not easily
29 done when tracing is enabled only once a problem has been
30 reported.

31

32 Furthermore, some ATM/kiosk vendors provide a limited
33 amount of non-volatile RAM. When this is provided, the
34 trace control writes the most recent trace information to
35 this RAM in a ring buffer fashion. As this is very

1 quick, it does not produce any performance problems.
2 However, if the ATM/kiosk freezes up or crashes, the RAM
3 contains the trace of what happened immediately before.

4

5 In addition to the traditional way that ActiveX® fires
6 events to the container, the device and self-service
7 controls are able to queue up events and return them one
8 by one when requested. This allows C++ applications to
9 be written in a procedural fashion rather than simply in
10 an event driven fashion. By queuing up these events and
11 delivering them to the application only on demand, the
12 system allows procedural code to be written and makes it
13 easier to develop and maintain the complex logic required
14 in self-service applications.

15

16 Important self-service controls are described below:

17

- 18 • Watchdog control: runs in a separate Windows NT®
19 process and reboots the ATM/kiosk if the application
20 crashes. This is achieved by regularly polling the
21 application to check that it is functioning correctly.
22 This control can also be used to daily reboot the
23 ATM/kiosk. The watchdog can monitor multiple
24 applications on a single ATM.
- 25 • System Escape control: used to reboot the ATM/kiosk.
26 Exits in a customisable manner. This control ensures
27 that cached data (eg in the DataCollect control and the
28 Trace control) is flushed to disk before rebooting.
- 29 • DataCollect control: allows application to collect raw
30 data for statistical purposes. It logs and timestamps
31 the various events. As with the Trace control, it logs
32 to memory and then stores on hard disk only when the
33 ATM/kiosk is idle due to the time required to write to
34 the hard disk. Storage by this control is of a fixed
35 size allocated at start-up and remaining constant

1 throughout operation. Storage is in the form of a ring
2 buffer. Typically, the collected data would be
3 exported to a remote location for analysis.

- 4 • Trace control: described above.
- 5 • Scratchpad control: described above.
- 6 • Supervisor application: run simultaneously as a
7 separate application. This means that on an ATM/kiosk
8 with a rear screen, the operator can interact with the
9 ATM/kiosk without taking the machine offline. It
10 allows the operator to access statistics etc. while the
11 machine is still being used. Alternatively, the
12 machine may be taken off-line for intrusive
13 maintenance. In this case, the supervisor application
14 provides an off-line mode with a limited subset of the
15 on-line features.

- 16 • Security control: described above.
- 17 • Registry control: allows Windows NT[®] registry to be
18 manipulated by the application.
- 19 • DirectoryTree control.
- 20 • Application Launcher control.
- 21 • INI file control: allows Windows[®] INI files to be read
22 from the browser.
- 23 • Timed FTP. This allows statistics files and trace files
24 to be sent via the FTP mechanism on a timed basis to an
25 offsite location. (eg daily or weekly).
- 26 • Key capture control: allows special Windows[®] key
27 combinations such as ctrl-alt-del and alt-tab to be
28 captured where a full PC keyboard is provided.
- 29 • Popup suppression control. Monitors and captures popup
30 windows originating from the operating system. This
31 makes it easier to allow software components from other
32 vendors to be used in self-service applications. Most
33 third-party software is not intended for self-service
34 applications and expects to be able to interact with

1 the user through popup windows. This is unacceptable
2 in a self-service environment where the main
3 application must have a complete monopoly over the user
4 dialog. This control alleviates this problem by
5 monitoring popups and rapidly executing a pre-
6 determined sequence of tasks, for example hiding the
7 popup and pressing the OK button.

- 8 • Global config file control. Allows configuration data
9 for ATM networks to be centrally held in a single
10 distributable file. Each ATM/kiosk can query this
11 control to retrieve the configuration data which is
12 specific for that ATM/kiosk. This allows variation
13 between individual ATMs/kiosks to be handled in a
14 global way.
- 15 • Telephony control. Allows modems and telephone handsets
16 to be integrated.
- 17 • SSMS control. Allows software to be downloaded and
18 installed in a controlled manner. This control checks
19 for installation failures and allows the system to
20 recover to a well defined state.
- 21 • Screensaver control. This control allows the
22 application to jump to a defined web page if the user
23 has been inactive for more than a pre-determined time.
- 24 • Multiple language control. This control allows the
25 language on a web page to be dynamically modified. It
26 does this by retrieving text strings and graphics from
27 a database on the kiosk. This means that the user may
28 change languages from any browser page - and therefore
29 at any stage of the application.
- 30 • Clock synch control. This allows the application to
31 synchronize its clock with a server clock, taking into
32 account possible differences in timezone between kiosk
33 and server and taking into account the possibility of
34 large timelags for communication between the kiosk and
35 the server.

1 Use of the self-service controls plus additional features
2 of the system and underlying operating system allow
3 ATMs/kiosks to be managed from a remote location. For
4 example, the system supports:

- 5 • Daily software downloads from a remote web server.
- 6 • Daily reboot and system check.
- 7 • Daily FTP of statistics data to a remote monitoring
8 station.
- 9 • Daily FTP of trace data to a remote monitoring system.
- 10 • Regular health checks of the kiosk (typically every 5
11 minutes).
- 12 • Sending a regular "heartbeat" message to a remote
13 monitoring station. Monitoring of this message allows
14 the fact that the device is continually functioning to
15 be monitored.
- 16 • Allowing direct secure access to the kiosk over a
17 network, for example the Internet, from a remote
18 location.
- 19 • Allowing software maintenance over a network, for
20 example the Internet, from a remote location.
- 21 • Allowing manual reboot of the kiosk over a network, for
22 example the Internet, from a remote location.

23
24 Although hardware is accessed via the WOSA XFS standard,
25 which assigns a different number to each command, the
26 controls have differently named methods and events
27 associated with each operation, making application
28 development easier. WOSA commands may typically generate
29 30-50 events. This wastes time for the application
30 developer and increases the possibilities of error. The
31 middleware reduces the set of possible outcomes to a
32 small number of clearly named completion events, making
33 it easier for the application developer to write reliable
34 code quickly. Outcomes which can only happen if there is

1 a bug in the application cause fatal errors to be
2 triggered.

3

4 The system automatically opens a WOSA XFS session when a
5 device control is first used; there is therefore no need
6 to manually call an Open method. WOSA sessions are
7 maintained between pages through the use of event
8 threads, described above.

9

10 All WOSA XFS methods require a timeout to be provided;
11 however, this is not appropriate or meaningful for the
12 majority of commands in this application. The middleware
13 requires a timeout to be supplied only where it is
14 meaningful to do so. WOSA also allows cancel commands to
15 be sent after any other command. Not all ATM functions
16 can really be cancelled and the middleware only provides
17 cancel commands where cancellation can actually be
18 achieved. The request IDs returned by WOSA for each
19 asynchronous operation are abstracted out by the
20 middleware. WOSA is accessed only by the middleware and
21 not directly by the application.

22

23 Clearly the preferred embodiment described above may
24 readily be adapted to operate with any operating system
25 or component system.

26

27 Further modifications and improvements may be
28 incorporated without departing from the scope of the
29 invention herein intended.

30

1 CLAIMS:

2

3 1. A method for providing transaction services wherein the
4 user of the transaction services interacts with a
5 computer-based transaction machine which has
6 transaction devices, said computer-based transaction
7 machine being controlled by one or more software
8 applications, said computer-based transaction machine
9 having an operating system which controls and receives
10 information from said transaction devices;
11 characterized in that the one or more software
12 applications interact with said transaction devices
13 through a programming interface of middleware software,
14 said middleware software interacting with the operating
15 system, wherein the services provided by the middleware
16 software depend on the capabilities of the transaction
17 devices but the programming interface does not depend
18 on the capabilities of the transaction devices.

19

20 2. A method for providing transaction services according
21 to Claim 1 wherein the transaction machine is selected
22 from a group which comprises automatic teller machines,
23 kiosks and electronic point of sale machines.

24

25 3. A method for providing transaction services according
26 to any preceding Claim wherein middleware software
27 comprises a series of transaction objects and controls
28 for standard device functions.

29

30 4. A method for providing transaction services according
31 to Claim 3 wherein transaction objects are independent
32 of the interface between the user and .../

M 09.00.00

1 24. A method for providing transaction services according
2 to any of Claims 19 to 23 wherein use of a web browser
3 allows conventional web sites to be displayed by the
4 computer-based transaction machine.

5

6 25. A method for providing transaction services according
7 to Claim 24 wherein middleware software allows or
8 disallows access to particular web sites according to a
9 rule database.

10

11 26. A method for providing transaction services according
12 to Claim 24 or Claim 25 wherein middleware software is
13 adapted to customise time-out of the display of
14 individual internet web sites.

15

16 27. A method for providing transaction services according
17 to any preceding Claim wherein the computer-based
18 transaction machine is adapted to allow the software
19 applications and middleware to be altered across a
20 network by an authority.

21

22 28. A method for providing transaction services according
23 to any preceding Claim wherein the transaction machine
24 can communicate information about their status to a
25 remote monitoring station across a network.

26

27 29. A computer-based transaction machine having transaction
28 devices; said computer-based transaction machine being
29 controlled by one or more software applications, said
30 computer-based transaction machine having an operating
31 system which controls and receives information from
32 said transaction devices; characterized in that the one

M 09.03.00
AB 34

1 or more software applications interact with said
2 transaction devices through a programming interface of
3 middleware software, said middleware software
4 interacting with the operating system, wherein the
5 services provided by the middleware software depend on
6 the capabilities of the transaction devices but the
7 programming interface does not depend on the
8 capabilities of the transaction devices.

9

10 30. A computer-based transaction machine according to Claim
11 29 wherein the transaction machine is selected from a
12 group which comprises automatic teller machines, kiosks
13 and electronic point of sale machines.

14

15 31. A computer-based transaction machine according to Claim
16 29 or Claim 30 wherein middleware software comprises a
17 series of transaction objects and controls for standard
18 device functions.

19

20 32. A computer-based transaction machine according to Claim
21 31 wherein transaction objects are independent of the
22 interface between the user and the transaction machine;
23 the interface between the user and the transaction
24 machine being customisable.

25

26 33. A computer-based transaction machine according to Claim
27 31 or Claim 32 wherein controls implement a
28 capabilities interface.

29

1 ABSTRACT

(Figure 2)

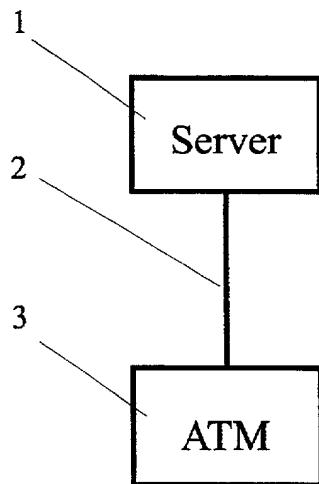
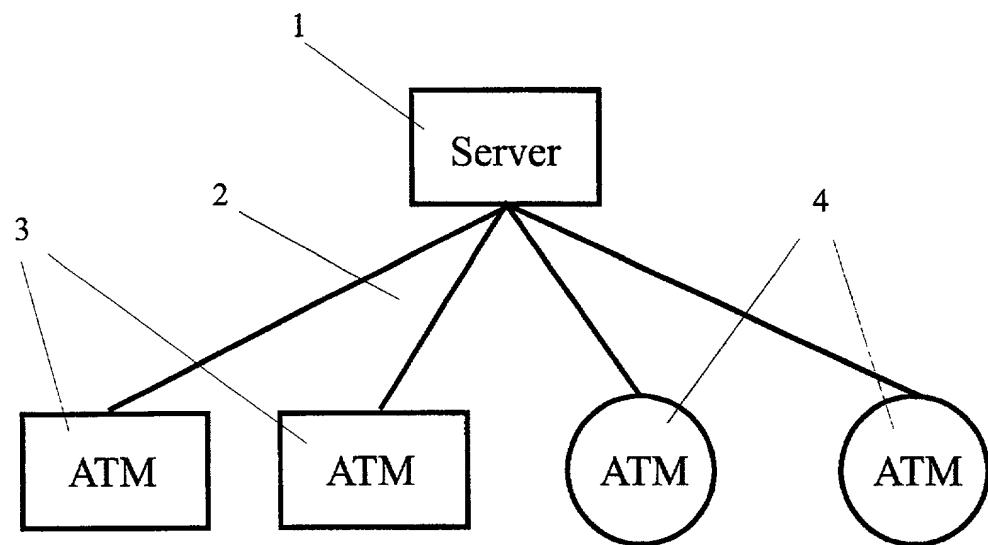
2

3 APPARATUS AND METHOD FOR PROVIDING TRANSACTION SERVICES.

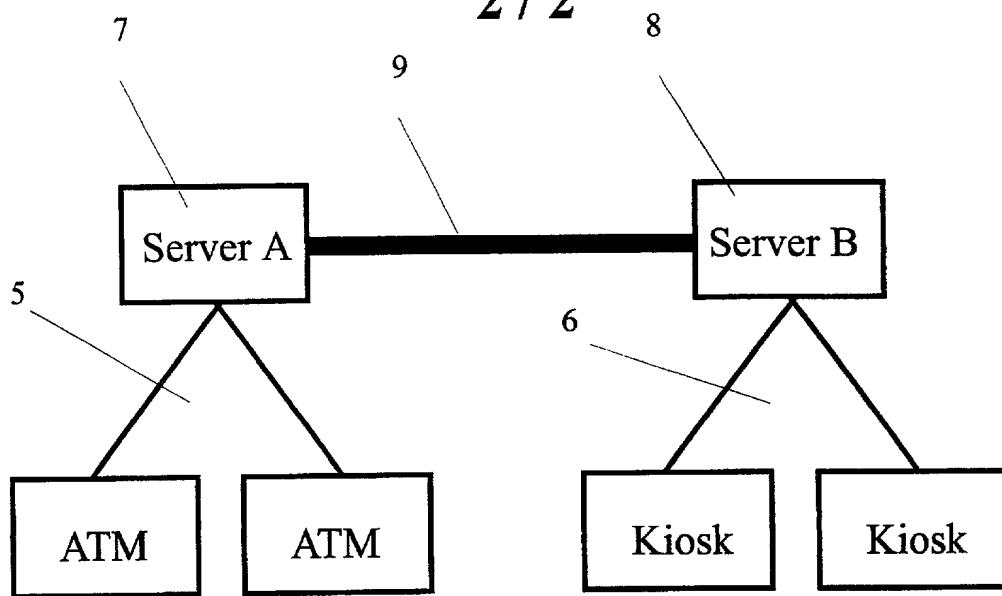
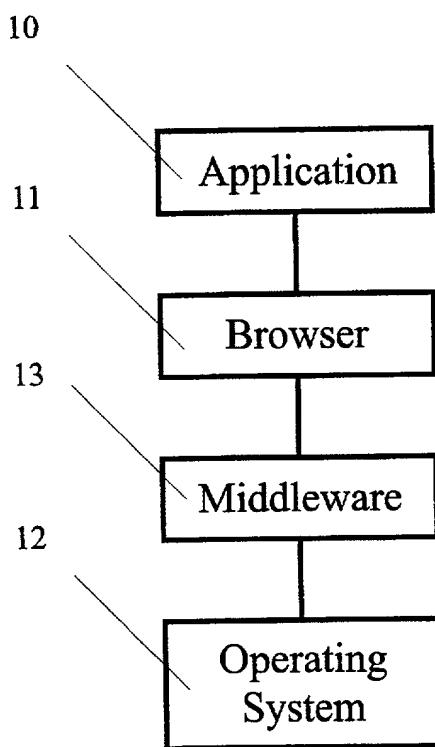
4

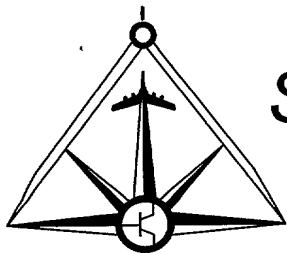
5 Apparatus and method for providing transaction services,
6 in particular a computer-based transaction machine, such
7 as an ATM, and a method for providing transaction
8 services using said transaction machine. One or more
9 software applications interact with middleware software
10 through functional interfaces that are hardware
11 independent but provide functionality which is
12 implemented in a manner adapted to the capabilities of
13 the particular hardware implementation. Objects provided
14 for standard transaction functions are independent of the
15 interface between the user and the transaction machine,
16 said interface being customisable. The resulting
17 transaction machines are typically combined into networks
18 and these networks may readily be combined to form an
19 Extranet.

1 / 2

**Figure 1****Figure 2**

2 / 2

**Figure 3****Figure 4**



Saltamar

Innovations

30 Fern lane
South Portland, ME 04106
U.S.A.
Phone: 1(207) 799-9733
Fax: 1(207) 799-3698
e-mail: saltamar@maine.rr.com

Declaration for Patent Application

(37 CFR 1.63)

(Foreign Agent Involved)

Application Number	
Filing Date	
Group Art Unit	
Examiner Name	

As a below named inventor, I hereby declare that:

My residence, post office address and citizenship are as stated below next to my name.

I believe I am the original, first and sole inventor (if only one name is listed below) or an original, first and joint inventor (if plural names are listed below) of the subject matter which is claimed and for which a patent is sought on the invention entitled:

Apparatus and Method for Providing Transaction Services

the specifications of which

 is attached hereto. was filed on Mar. 24, 1999 as United States Application Number or PCT Application Number PCT/GB99/00927 and was amended on Mar. 7 2000, and Sep. 22 2000 (if applicable).

I hereby state that I have reviewed and understand the contents of the above identified specification, including the claims, as amended by any amendment referred to above.

I acknowledge the duty to disclose information which is material to patentability as defined in 37 CFR § 1.56.

I hereby claim foreign priority benefits under 35 U.S.C. § 119(a)-(d) or § 365(b) of any foreign application(s) for patent or inventor's certificate, or § 365(a) of any PCT International application which designated at least one country other than the United States, listed below and have also identified below, by checking the box, any foreign application for patent or inventor's certificate, or PCT International application having a filing date before that of the application on which priority is claimed.

Priority NOT claimed Copy attached

<u>PCT/GB00927</u> (Number) 9806843.0	<u>WO</u> (Country) UK	<u>24/3/1999</u> (DD/MM/YYYY Filed) 24/3/1998	<input type="checkbox"/>	<input type="checkbox"/>
(Number)	(Country)	(DD/MM/YYYY Filed)	<input type="checkbox"/>	<input type="checkbox"/>

 Additional foreign application numbers are listed on a supplemental priority data sheet

I hereby claim the benefit under 35 U.S.C. § 120 of any United States application(s), or § 365 (c) of any PCT International application designating the United States, listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in the prior United States or PCT International application in the manner provided by the first paragraph of 35 U.S.C. § 112, I acknowledge the duty to disclose information which is material to patentability as defined in 37 CFR § 1.56 which became available between the filing date of the prior application and the national or PCT International filing date of this application.

(Application Number) (Filing Date) (Status)

(Application Number) (Filing Date) (Status) Additional US or PCT international application numbers are listed on a supplemental priority data sheet

I hereby claim the benefit under 35 U.S.C. 119(e) of any United States provisional applications listed below.

Application Number(s)	Filing Date	<input type="checkbox"/> Additional provisional application numbers are listed on supplemental priority data sheet attached hereto

Power of Attorney and instruction Authorization

I hereby appoint Mr. Shalom Wertsberger of the firm Saltamar Innovations, Reg. No 43,359 to prosecute this application and to transact all business in the Patent and Trademark Office connected therewith.

I hereby authorize Shalom Wertsberger to accept and follow instructions from Mr. David Kennedy, Esq., as to any action to be taken in the Patent and Trademark Office regarding this application without direct communication between the U.S. attorney or agent and the undersigned. In the event of a change in the persons from whom instructions may be taken, the U.S. agent named herein will be so notified by the undersigned by registered mail directed to:

(1)

Shalom Wertsberger
Saltamar Innovations
30 Fern Lane
South Portland, ME 04106

Address all telephone calls to: (207) 799-9733

Address all facsimile inquiries to: (207) 799-3698

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief am believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issuing thereon.

Full name of sole or first named inventor (given name, family name) Aravinda Korala

Ras. Alaa.

(Inventor Signature)

20 Oct 2000

(MM/DD/YYYY Date Executed)

Edinburgh, UK

G BX

(Residence [town, state])

FR

(Citizenship)

John Cotton Building, Edinburgh, EH7 5RA, United Kingdom

(Post office Address)

Full name of second or joint named inventor (given name, family name) _____

(Inventor Signature)

(MM/DD/YYYY Date Executed)

(Residence)

(Citizenship)

(Post office Address)

Additional inventors are being named on _____ sheets attached hereto